Mark Wolfe's Blog

Resume

Using AWS DeepRacer for ROS development

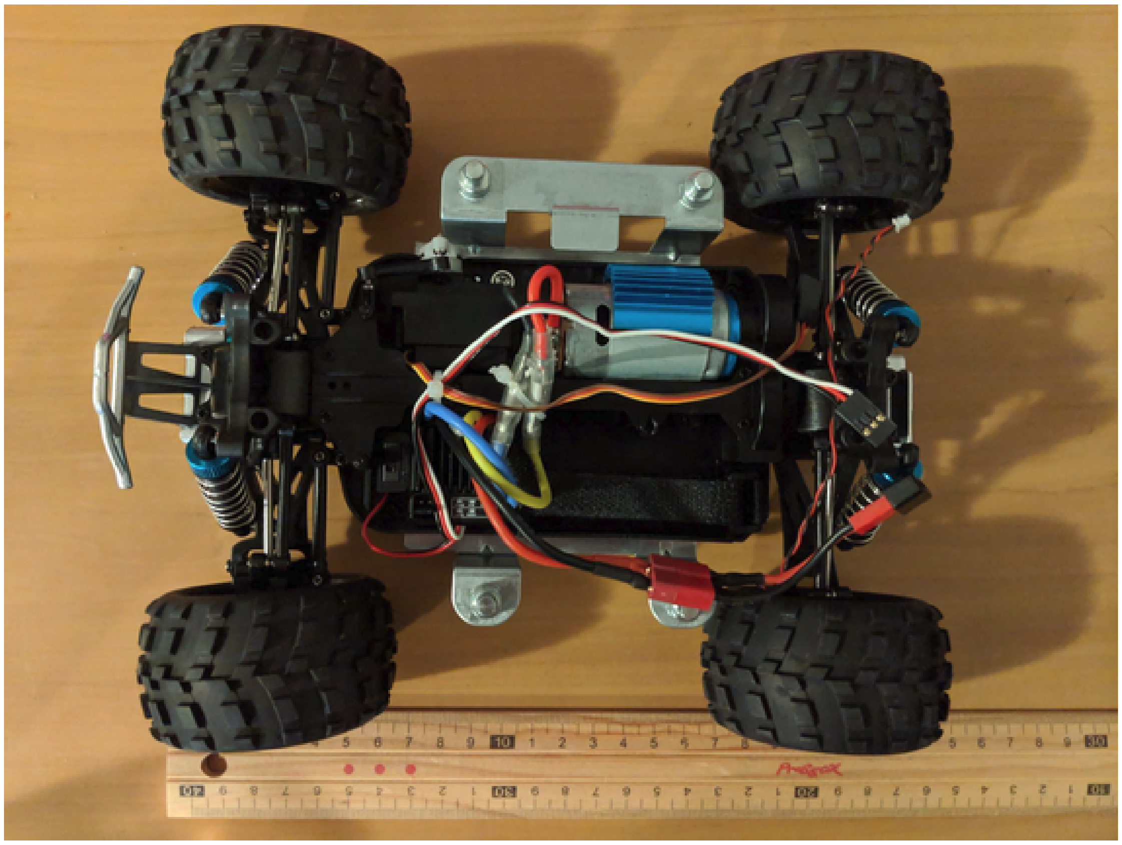
So the DeepRacer was released with much fanfare by Amazon Web Services (AWS) at this years Reinvent conference in Las Vegas. This combines an off the shelf radio control (RC) car chassis, with an intel based compute module which has been configured to control the throttle, steering of the car. DeepRacer is powered by Robot Operating System (ROS) as a framework which is used for the internal control systems making it a very interesting device for anyone getting started in robotics.

In this post I will show how to use the DeepRacer for ROS development.

So why would I use DeepRacer for ROS development?

1. Intel based so very easy to build and upload binaries / projects.
2. It works out of the box and is very well designed.
3. It will retail for 249 USD in march, given it’s high quality components it will be very useful.
4. It is based on Ubuntu 16.04.3, and ROS Kinetic which is a good solid starting point for ROS Development.

Below is a picture the RC car chassis which is a really nice size.



Logging into your DeepRacer

Logging into the DeepRacer for the first time is very similar to the DeepLens product, plug the compute module into a monitor using a HDMI cable and a USB Keyboard and mouse, then reset the users password to some value which satisfies the complexity requirements. I believe it is at least 8 characters, one upper, lower, number and a symbol.

Note: The username is `deepracer`.

Preparing your DeepRacer

In my case I also disabled x11, and enabled SSH.

To permit ssh through the firewall configured on the device I ran, the server was already installed and started.

```
sudo ufw allow ssh
```

To disable the desktop environment on the next restart run the following.

```
sudo systemctl disable lightdm.service
```

Note: I did this while the screen and keyboard where plugged in, then tested I could ssh to the DeepRacer from my laptop before restarting.

Now that you have ssh'd into the DeepRacer using your login details, and the `TAG_HOSTNAME` is on a tag on the bottom of the car.

```
ssh deepracer@TAG_HOSTNAME
```

Before you starting to mess around on the host you probably want to disable ufw to enable access to all the services from your laptop/desktop.

```
sudo ufw disable
```

Once you have disabled the firewall your should be able to access the video stream from ROS `web_video_server` using http://TAG_HOSTNAME.local:8080/stream_viewer?topic=/video_mjpeg where the `TAG_HOSTNAME` is on a tag on the bottom of the car. For more information on this service see http://wiki.ros.org/web_video_server.

Disable ROS Services

We are going to disable services which relate to the DeepRacer service, including the updates service so we can just use the control and media systems.

This is as simple as replacing `/opt/aws/deepracer/share/deepracer_launcher/launch/deepracer.launch` with content as follows, this simply comments out some of the AWS supplied services.

```
<?xml version="1.0"?>
<launch>
    <node pkg="web_video_server" type="web_vic
        <param name="ros_threads" value="1" />
    </node>
    <node name="servo_node" pkg="servo_pkg" ty
    <node name="media_engine" pkg="media_pkg"

    <!-- <node name="inference_engine" pkg="in
    <!-- <node name="inference_probability" pl
    <!-- <node name="model_optimizer" pkg="in

    <node name="control_node" pkg="ctrl_pkg" t
    <!-- <node name="navigation_node" pkg="ctr
    <!-- <node name="software_update" pkg="sof
    <!-- <node name="webserver" pkg="webserver
</launch>
```

Now restart the DeepRacer service.

```
$ sudo systemctl restart deepracer-core.servic
```

Is this thing on?

To test whether or not we can still drive the DeepRacer around, we will explore then interface with the control node provided.

Now we load the ROS environment, using the AWS `setup.bash`, this will populate variables holding names and paths for services, if you have never used ROS before you may want to run through some of the Tutorials.

```
$ source /opt/aws/deepracer/setup.bash
```

Now we should just have the services we need to start working with ROS.

```
$ rosnode list
/control_node
/media_engine
/rosout
/servo_node
/web_video_server
```

Lets look at the topics which are now accepting messages.

```
$ rostopic list
/auto_drive
/calibration_drive
/manual_drive
```

```
/rosout
/rosout_agg
/video_mjpeg
```

Now we are interested in `/manual_drive` to test out the throttle and steering.

```
$ rostopic info /manual_drive
Type: ctrl_pkg/ServoCtrlMsg

Publishers: None

Subscribers:
 * /control_node (http://amss-n4lp:37837/)
```

So we need to post a message to this topic, but we need to know the format so lets print it’s structure.

```
$ rosmsg info ctrl_pkg/ServoCtrlMsg
float32 angle
float32 throttle
```

Now while the DeepRacer is off my desk, to stop it racing off, I run the following command which should trigger a throttle change. Note that the limit for this value is `0.7` by default, and `0` will stop it.

```
rostopic pub -1 /manual_drive ctrl_pkg/ServoCt
```

Now we can stop the throttle by running.

```
rostopic pub -1 /manual_drive ctrl_pkg/ServoCt
```

Likewise we can turn the DeepRacer to the left using the first value.

```
rostopic pub -1 /manual_drive ctrl_pkg/ServoCt
```

And to the right.

```
rostopic pub -1 /manual_drive ctrl_pkg/ServoCt
```

Then back to the center.

```
rostopic pub -1 /manual_drive ctrl_pkg/ServoCt
```

I am really impressed with the DeepRacer so far, it is a great platform to start working with ROS at a great price, hopefully it spurs a whole raft of great robotics projects in the future. I would also love to see more detail, and hopefully source code for the services in this product as they seem to be really well thought out and could most certainly provide a spring board for future innovation.

In my next post we will write a ROS node which will use these services to drive the DeepRacer.

2 Comments Mark Wolfe's Blog 1 Login

Recommend Tweet Share Sort by Best

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

RyeBrye • 4 months ago

One thing that I've not had time to do yet is get something to let me use an external Bluetooth controller (the car doesn't seem to have Bluetooth, but using a be USB dongle shouldn't be hard) - maybe a dualshock 4 controller - and then mapping the analog controls from the controller over to drive the car.

I was thinking that doing that as a simple python app sending controls to the existing manual drive control node would be the best way to go about it - but I don't know much about ROS other than what I figured out by tinkering on this car so far.

• Reply • Share ›

Alex Krolicki ➦ RyeBrye • a month ago

Any updates on using a bluetooth controller? Very interested as I am attempting the same thing.